VeriSilicon GSMC 0.15μm High-Speed Synchronous Single-Port/Dual-Port SRAM Compiler User's Guide

Trademark Acknowledgments

VeriSilicon & the VeriSilicon logo are the trademarks of VeriSilicon Microelectronics (Shanghai) Co., Ltd.

All other products and company names mentioned may be the trademarks of their respective owners.

© 2002 VeriSilicon Microelectronics (Shanghai) Co., Ltd. All rights reserved.

Printed in P.R.China.

VeriSilicon Microelectronics (Shanghai) Co., Ltd. reserves all its copy rights and other intellectual property rights, ownership, powers, benefits and rights arising or to arise from this manual. All or part of the contents of this manual may be changed by VeriSilicon Microelectronics (Shanghai) Co., Ltd. without notice at any time for any reason, including but not limited to improvement of the product relating hereto.

VeriSilicon Microelectronics (Shanghai) Co., Ltd. shall not undertake or assume any obligation, responsibility or liability arising out of or in respect of the application or use of the product described herein, except for reasonable, careful and normal uses.

Nothing, whether in whole or in part, within this manual can be reproduced, duplicated, copied, changed or disposed of in any form or by any means without prior written consent by VeriSilicon Microelectronics (Shanghai) Co., Ltd..

VeriSilicon Microelectronics (Shanghai) Co., Ltd.

3F, Building 1, No.200, Zhangheng Road, Zhangjiang Hi-Tech Park, Pudong New Area, Shanghai 201204, P. R. China

Tel : +86-21-5131-1118

Fax : +86-21-5131-1119

Web : http://www.verisilicon.com

Contents

Chapter 1 Introduction 4

1.1 Compiler Description....4
1.2 Features....4
1.3 Operating Conditions....4
1.4 Write Mask Description....5
1.5 Pin Descriptions....5
1.6 Parameter Range....6
1.7 SRAM Floor Plan....7

Chapter 2 Timing Diagram 8

2.1 Timing Specifications for Dual-Port SRAM....8
Clock Contention Diagram....8
Timing Diagrams....9
Timing Parameters....10
Power Parameters....11
2.2 Timing Specifications for Single-Port SRAM....12
Timing Diagrams....12
Timing Parameters....13
Power Parameters....13

Chapter 3 Using the SRAM Compiler 15

3.1 System Requirement....15
3.2 Software Environment....15
3.3 Installing SRAM Compiler....15
3.4 Inputs and Outputs....16
3.5 Getting Started....17
Using Shell Commands....17
Using Graphical User Interface (GUI)....18
3.6 Generating the Outputs....21

chapter 1 Introduction

1.1 Compiler Description

VeriSilicon GSMC 0.15μm High-Speed Synchronous Single-Port/Dual-Port SRAM compiler optimized for Grace Semiconductor Manufacturing Corporation (GSMC) 0.15μm Logic 1P7M Salicide 1.5/3.3V process can flexibly generate memory blocks by a friendly GUI or shell commands.

The compiler supports comprehensive range of word and bits. While satisfying speed and power requirements, it was optimized for area efficiency.

VeriSilicon GSMC Synchronous Single-Port/Dual-Port SRAM compiler uses four layers within the blocks and supports metal 4, 5, 6 or 7 as the top metal. Dummy bit cells are designed in with the intention to enhance reliability.

1.2 Features

- Single or Dual Read/Write Ports
- High Density
- High Speed
- Size Sensitive Self-time Delay for Fast Access Time
- Automatic Power Down
- Tri-state Output
- Write mask function

1.3 Operating Conditions

The following table gives the recommended operating conditions for memory blocks generated by SRAM compiler:

Recommended Operating Conditions

Operating Conditions	Parameter	Rating
Supply Voltage	1.5V	1.35V to 1.65V
Temperature	25°C	0°C to 125°C

1.4 Write Mask Description

The SRAM compiler supports write mask selection. When write mask are selected to off, the SRAM works at normal mode without write mask. When write mask are selected to on, the SRAM compiler works at write mask mode.

If Single-Port SRAM works at normal mode, all of the data will be written into the memory when CEN is low and WE is high. If Single-Port SRAM works at write mask mode, the data input bus DIN[i:0] will be partitioned into several groups which enabled by write enable bus WE[j:0], Each group data inputs are enabled by corresponding write enable pin separately. For example, if a single port SRAM with 16 bits is selected to write mask mode, and the write mask group is 4, then write enable bus will be WE[3:0], and DIN[3:0] will be controlled by WE[0], DIN[7:4] will be controlled by WE[2], DIN[11:8] will be controlled by WE[2], DIN[15:12] will be controlled by WE[3] separately.

If Dual-Port SRAM works at normal mode, all of the data in port A/B will be written into the memory when CENA/CENB is low and WEA/WEB is high. If Dual-Port SRAM works at write mask mode, the data input bus DAIN[i:0]/ DBIN[i:0] will be partitioned into several groups which enabled by write enable bus WEA[j:0]/WEB[j:0], each group data inputs are enabled by corresponding write enable pin separately. For example, if a dual port SRAM with 16 bits is selected to write mask mode, and the write mask group is 4, then write enable bus will be WEA[3:0]/WEB[3:0], and DAIN[3:0]/DBIN[3:0] will be controlled by WEA[0]/WEB[0], DAIN[7:0]/DBIN[7:4] will be controlled by WEA[2]/WEB[2], DAIN[11:8]/DBIN[11:8] will be controlled by WEA[2]/WEB[2], DAIN[15:12]/DBIN[15:12] will be controlled by WEA[3]/ WEB[3] separately.

1.5 Pin Descriptions

The following table gives detailed information of pin descriptions for single-port SRAM (Bus index descending ordered):

Pin	Description
DOUT[i]	Data output
AD[i]	Address input
CEN	Chip enable input, low enable
CLK	CLK input, positive edge active
DIN[i]	Data input
OE	Output enable, high enable
WE	Write or Read control input, Write high enable, Read low enable

The following table gives detailed information of pin descriptions for

dual-port SRAM (Bus index descending ordered):

Pin	Description
DAOUT[i]	A port data output
DBOUT[i]	B port data output
ADA[i]	A port address input
ADB[i]	B port address input
CENA	A port chip enable input, low enable
CENB	B port chip enable input, low enable
CLKA	A port CLK input, positive edge active
CLKB	B port CLK input, positive edge active
DAIN[i]	A port data input
DBIN[i]	B port data input
OEA	A port output enable, high enable
OEB	B port output enable, high enable
WEA	A port Write or Read control input, Write high enable, Read low enable
WEB	B port Write or Read control input, Write high enable, Read low enable

1.6 Parameter Range

<table>
<tr><th>Parameter</th><th colspan="3">Range</th></tr>
<tr><td>Memory Array Range</td><td colspan="3">32 to 512k Bits</td></tr>
<tr><td>Data Width</td><td colspan="3">Mux4: 2 to 256 Bits, Increments of 1
Mux8: 2 to 128 Bits, Increments of 1
Mux16: 2 to 64 Bits, Increments of 1</td></tr>
<tr><td>SRAM Address Depth</td><td colspan="3">Mux4: 16 to 2048 Words, Increments of (2 X 4)
Mux8: 32 to 4096 Words, Increments of (2 X 8)
Mux16: 64 to 8192 Words, Increments of (2 X 16)</td></tr>
<tr><td rowspan="5">Column Mux</td><td colspan="3">The following list the changes of width and height when column mux is set to different values. Suppose column mux is 16, the width and height is a standard.</td></tr>
<tr><td></td><td>Width</td><td>Height</td></tr>
<tr><td>4</td><td>1/4</td><td>4</td></tr>
<tr><td>8</td><td>1/2</td><td>2</td></tr>
<tr><td>16</td><td>1</td><td>1</td></tr>
<tr><td>Top Metal</td><td colspan="3">m4, m5, m6 or m7</td></tr>
<tr><td>Output Drive Strength</td><td colspan="3">The same drive as INVHD8X cell in VeriSilicon GSMC 0.15μm High-Density Standard Cell Library</td></tr>
</table>

1.7 SRAM Floor Plan

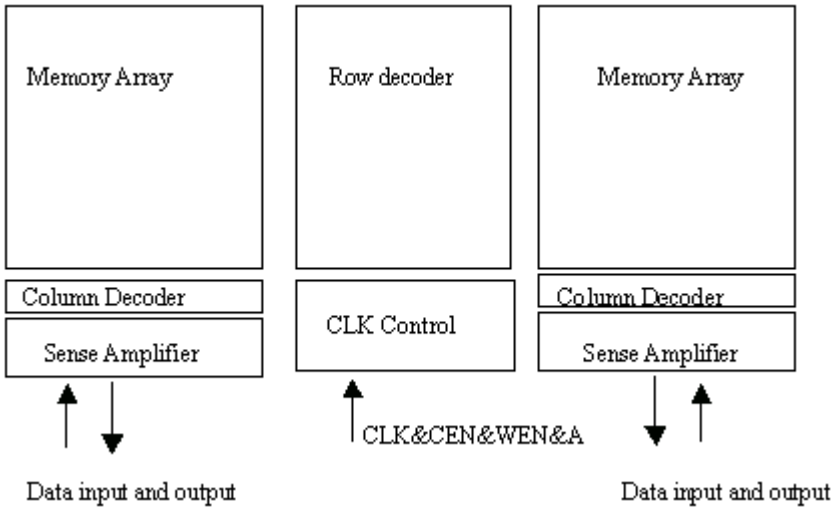

Fig.1 SRAM Floor Plan

chapter 2 Timing Diagram

2.1 Timing Specifications for Dual-Port SRAM

This section specifies the timing specifications of the synchronous dual-port SRAM.

Clock Contention Diagram

When port A and B access the same address, the clock contention will happen. The detail clock contention diagrams are shown as follows:

t_{cc} is clock collision time reported in datasheet.

First read one port then write the other

If t > t_{cc} (see Fig.2) write and read are both OK.

If t < t_{cc} (see Fig.2) write is OK but read fails.

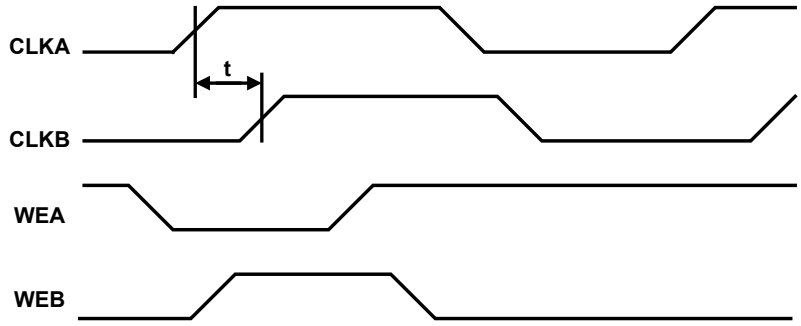

Fig.2 First read then write

First write one port then read the other

If t > t_{cc} (see Fig.3) write and read are both OK.

If t < t_{cc} (see Fig.3) write is OK but read fails.

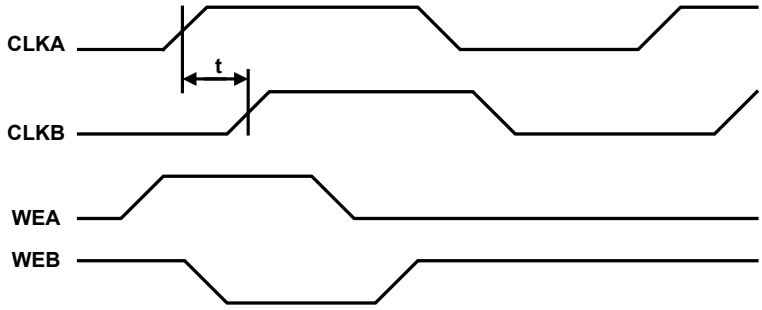

Fig.3 First write then read

First write one port then write from the other

If t > t_{cc} (see Fig.4) both write are OK.

If t < t_{cc} (see Fig.4) both write fail.

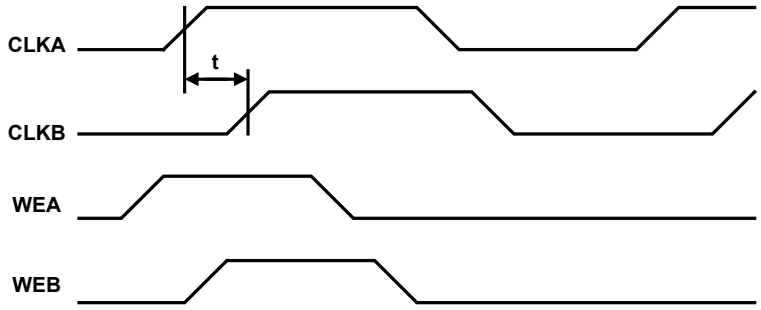

Fig.4 First write then write

First read one port then read the other

If t > t_{cc} (see Fig.5) both read are OK.

If t < t_{cc} (see Fig.5) both read are OK.

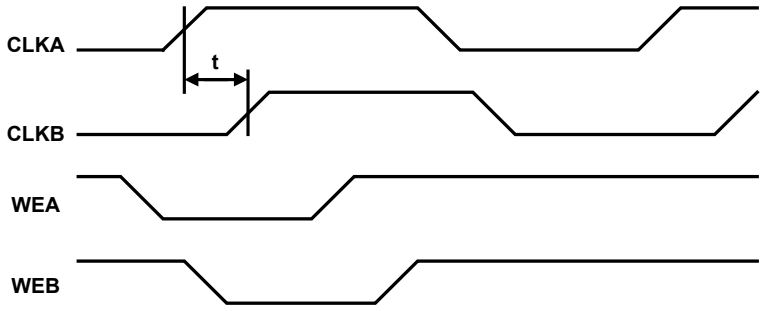

Fig.5 First read then read

Timing Diagrams

The synchronous SRAM write, read and output enable timing diagrams are shown as follows:

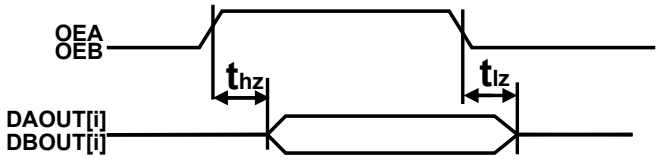

Fig.6 Output-Enable Function Timing

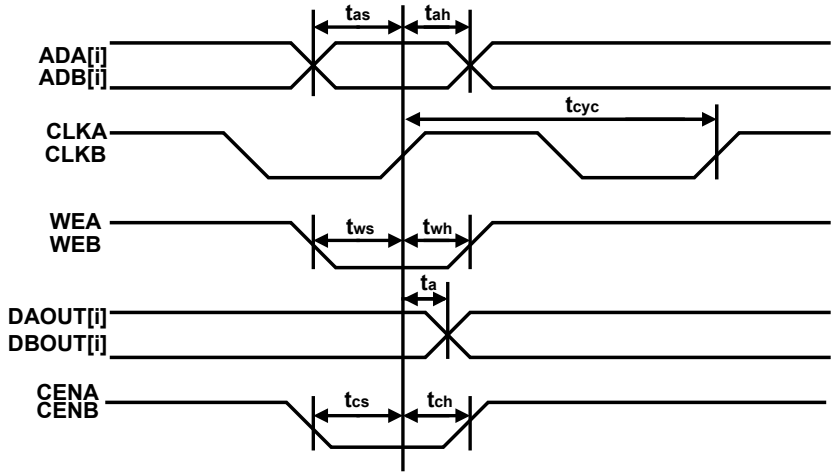

Fig.7 Read-Function Timing

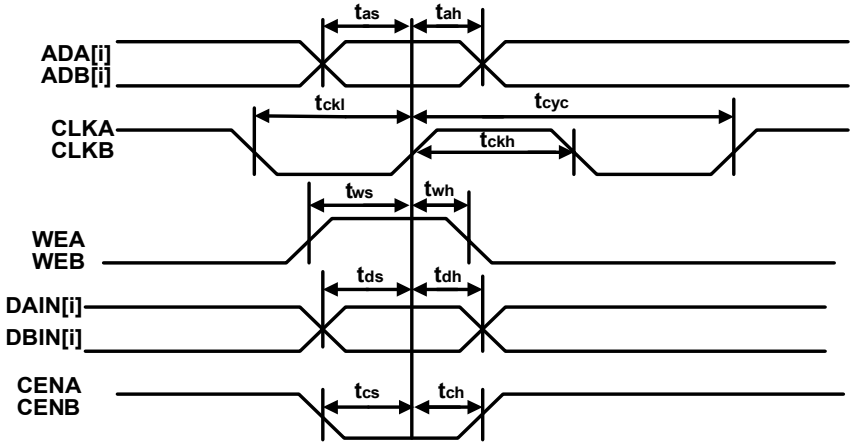

Fig.8 Write-Function Timing

Timing Parameters

The following table specifies the timing parameters in the datasheet generated by the SRAM compiler:

Timing Parameters

Parameter	Symbol
Cycle time	tcyc

Access time*	ta
Write enable setup time	tws
Write enable hold time	twh
Address setup time	tas
Address hold time	tah
Data setup time	tds
Data hold time	tdh
Chip enable setup time	tcs
Chip enable hold time	tch
Output enable to hi-Z time	tlz
Output enable active time*	thz
Clock collision time	tcc

* Parameters have a load dependence

Power Parameters

The following table specifies the power parameters in the datasheet generated by the SRAM compiler:

Power Parameters

Parameter	Symbol
Average current	Iavg
Peak current	Ipeak

i_{avg} is the average current in A/100MHz unit. The average current in the datasheet is achieved under below assumption:

1. Input net transition is 0.2ns.
2. Output port capacitance is 0pF.
3. 50% cycle read and 50% cycle write. That is:

$i_{avg} = (i_{avg}(read) + i_{avg}(write))/2$

Where, iavg(read) is the average current of read. And, iavg(write) is the average current of write.

The total average current of the memory than can be estimated according following equation:

$I_{avg} = i_{avg} * F + 1/2 * C * V * f * N$

Where,

I_{avg}: the total average current of the memory. (A)

F: the frequency of clock. (100MHz)

C: the average capacitance of output port. (F)

V: the voltage supply. (V)

f: the frequency of output port. (Hz)

N: switched ports number.

i_{peak} is the peak current of memory during operation in A unit.

2.2 Timing Specifications for Single-Port SRAM

This section specifies the timing parameters of the synchronous single-port SRAM.

Timing Diagrams

The synchronous SRAM write, read and output enable timing diagrams are shown as follows:

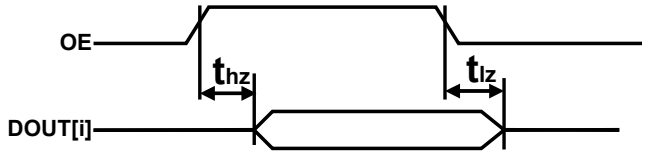

Fig.9 Output-Enable Function Timing

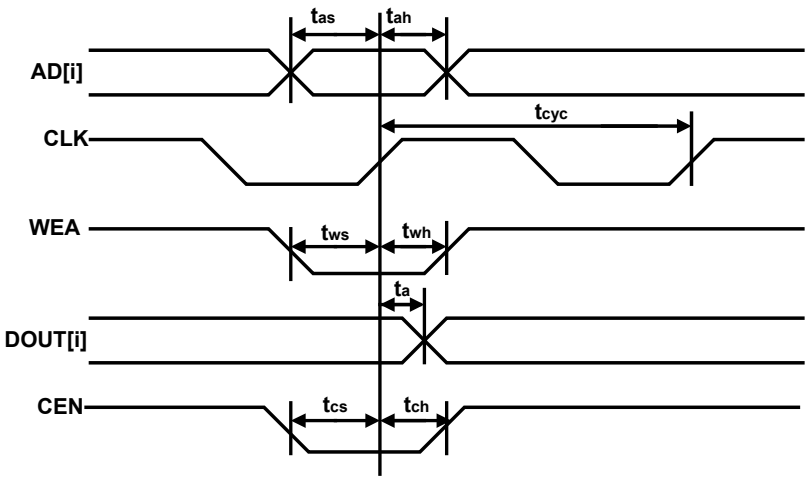

Fig.10 Read-Function Timing

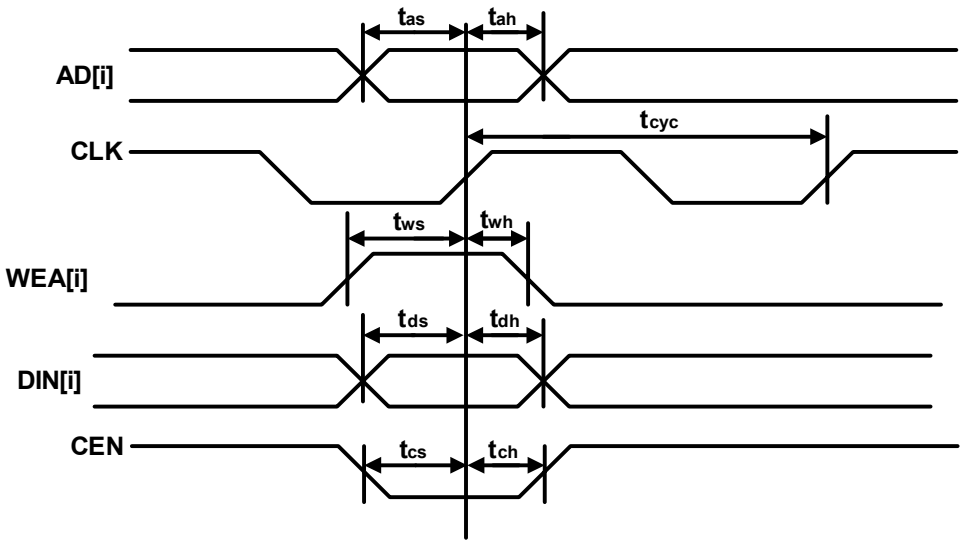

Fig.11 Write-Function Timing

Timing Parameters

The following table specifies the timing parameters in the datasheet generated by the SRAM compiler:

Timing Parameters

Parameter	Symbol
Cycle time	tcyc
Access time*	ta
Write enable setup	tws
Write enable hold	twh
Address setup	tas
Address hold	tah
Data setup	tds
Data hold	tdh
Chip enable setup	tcs
Chip enable hold	tch
Output enable to hi-Z	tlz
Output enable active*	thz

* Parameters have load dependence.

Power Parameters

The following table specifies the timing parameters in the datasheet generated by the SRAM compiler:

Power Parameters

Parameter	Symbol
Average current	Iavg
Peak current	Ipeak

i_{avg} is the average current in A/100MHz unit. The average current in the datasheet is achieved under below assumption:

3. Input net transition is 0.2ns.
4. Output port capacitance is 0pF.
3. 50% cycle read and 50% cycle write. That is:

$i_{avg} = (i_{avg}(read) + i_{avg}(write))/2$

Where, iavg(read) is the average current of read. And, iavg(write) is the average current of write.

The total average current of the memory than can be estimated according following equation:

$I_{avg} = i_{avg} * F + 1/2 * C * V * f * N$

Where,

I_{avg}: the total average current of the memory. (A)

F: the frequency of clock. (100MHz)

C: the average capacitance of output port. (F)

V: the voltage supply. (V)

f: the frequency of output port. (Hz)

N: switched ports number.

i_{peak} is the peak current of memory during operation in A unit.

chapter 3 Using the SRAM Compiler

3.1 System Requirement

Before installation, make sure that the following minimum host configuration is available:

- Sun Microsystem's Solaris7

3.2 Software Environment

The SRAM compiler will run on UNIX and X-Window and its GUI was developed with motif.

3.3 Installing SRAM Compiler

Please follow the following instruction to install SRAM compiler step by step:

1. Create an installation directory where you wish to install the SRAM compiler.

> ***NOTE***
> *<install_dir> will stand for the directory you have created for installation hereafter.*

2. cd <install_dir>
3. gunzip < <release_compressed_file> | tar xvf -
4. Copy .vsmcrc file to the home directory.
5. Add the following to .cshrc file.

   source ~/.vsmcrc
6. Modify .vsmcrc file as the following and source it:

   setenv VERISILICON_MC_DIR <install_dir>

After successful installation, the following directory will be created under <*install_dir*>:

gsmc_rasp.15

This directory contains the technology files and library files of the Single-Port

SRAM compiler.

gsmc_radp.15

This directory contains the technology files and library files of the Dual-Port SRAM compiler.

The following table specifies the executable file names in the installation directory and their descriptions:

Name	Description
MC	The executable file of SRAM compiler.

NOTE

Be sure not to edit any files in <install_dir> *directory*.

3.4 Inputs and Outputs

The SRAM compiler allows users to define the following parameters for a specific SRAM block:

Library

Running Directory

Block Name

Number of Words

Number of Bits

Ring Width

Frequency (MHz) (no use now)

Write Mask Number

Multiplexer Width

Horizontal Ring Layer

Vertical Ring Layer

Top Metal Layer

The SRAM compiler generates the following outputs for further use.

GDSII Layout File (GDSII format)

LVS Netlist (CDL format)

Verilog Model Code

VHDL Model Code

TLF Model

Synopsys Model

Datasheet

LEF view

Antenna LEF view

Antenna CLF model

After the above parameters input, users should wait a few minutes for the outputs to be generated by the SRAM compiler automatically.

3.5 Getting Started

There are two ways to start SRAM compiler as follows:

Using Shell Commands

Users can specify the parameters in the SRAM compiler commands with options to launch SRAM compiler. Enter the following commands to launch the SRAM compiler directly from the shell:

% cd <running_dir>

% MC [options with parameters]

The <*running_dir*> is the directory which the SRAM compiler run in. All the outputs will be generated in the specified directory.

Be sure that the running directory < *running_dir*> should not be the same as the installation directory <*install_dir*>.

All the options with parameters for the SRAM compiler are listed as follows:

-lib $lib_dir

-outdir $run_dir

-block $mem_name

-wordsnumber $memlength

-bitsnumber $datawidth

-ringwidth $ringwidth

-muxwidth $varMuxWidth

-vlayer $varVLayer

-hlayer $varHLayer

-frequency $frequency (no use now)

-topmetal $topmetal

-writemask $writemask

-area y/n

Please see section **Parameters under GUI and Shell Commands** for

details about parameters.

Example:

MC –lib gsmc_radp.15 -outdir . -block MCDP1024X16 -wordsnumber 1024 -bitsnumber 16 -ringwidth 5 -muxwidth 16 -vlayer m4 -hlayer m3 -topmetal m4 -frequency 100 –writemask 0

This command will generate a dual-port SRAM name of MCDP1024X16, with 1024 words, 16 bits, 16 column multiplexer width, frequency 100, metal layer 4 as the vertical ring layer, metal layer 3 as the horizontal ring layer, m4 as the top metal layer, 0 as write mask.

NOTE

$writemask default value is 0, and $writemask=0 is equal to $writemask=$datawidth.

Using Graphical User Interface (GUI)

We provide a friendly GUI to enable the users to configure parameters and generate all the outputs in the directory specified. Using the GUI from the shell, input the commands as follows:

% cd <running_dir>

% MC

Click on the *browse* button to select the Single-Port or Dual-Port SRAM's library, then the GUI for the SRAM compiler will appear on your screen as follows:

Fig.1 the SRAM Compiler GUI

Fill content in the blank for each option and click the proper button, you will get your results.

Parameters under GUI and Shell Commands

This section specifies detailed descriptions of the parameters of the SRAM compiler and their corresponding default values:

Parameter under Shell Commands	Parameter under GUI	Description	Default Value	Range
-lib	**Library**	Specify the library directory used by the SRAM compiler. You can click on the *browse* button to find the valid library you have installed.	The library used in last time. (it is record in <install_dir>/mc.ini)	
-rundir	**Running Directory**	Specify the output directory of the SRAM compiler. It can be any valid path name supported by the system.	.	

-block	**Block Name**	Specify the block name. It can include any alphanumeric value and must be unique to avoid name conflicts for blocks within the same library. It is recommended that a block name is no more than 16 characters, for we will identify two blocks by their first 16 letters.	RASP2048X16M16	
-wordsnumber	**Number of Words**	Specify the number of words in the block.	2048	32 to 8192
-bitsnumber	**Number of Bits**	Specify the number of bits in the block.	16	2 to128
-ringwidth	**Ring Width**	Specify the ring width of the block in um. The minimum is 2. The designer must decide the ring width according to the power analysis. The average current reported in the datasheet supposes 50% of address and data switch, 50% write and read operations, and one of dual ports active.	5	
-frequency	**Frequency**	Specify the frequency of the clock of the block in MHz. It dose not affect anything now. and it is just reserved for future usage.	100MHz	
-muxwidth	**Multiplexer Width**	Specify the column multiplexer width. There are three buttons for your choice: 4, 8, or 16. When this option is set to different values, the width and height of the block will change correspondingly. For detailed information, please refer to Parameter Range section on page 6.	16	
-vlayer	**Vertical Ring Layer**	Specify which metal layer will be the vertical ring layer. There are four or your choice: m1, m2, m3, or m4.	m4	M1-M4

-hlayer	**Horizontal Ring Layer**	Specify which metal layer will be the horizontal ring layer. There are four for your choice: m1, m2, m3, or m4.	m3	M1-M4
-topmetal	**Top Metal Layer**	Specify the top metal layer. It can be m4, m5 or m6. m5 or m6 will be stand for all other higher layers.	m4	M4-M6
-writemask	**Write Mask Number**	Specify how many bits will be controlled by one 'we'.	0	0 to number of bits
-area	**y / n**	Once "–area y" is used, the compiler will only generate a width X height report. Default is n.	n	y or n

NOTE

The horizontal and vertical ring layers must be set to two different metal layers, for example, m1 and m2. When a value is selected, the vertical ring layer is automatically set to a valid value.

3.6 Generating the Outputs

When you click on the *Default* button in the GUI, the SRAM compiler will automatically load the default parameters of the SRAM and generate the SRAM based on the default parameters. Thus the SRAM compiler will be initialized to their default values.

To generate the outputs, click on *Generate* button in the GUI. All the outputs will be generated according to the parameters you set and place in the running directory <*running_dir*> users specify.

The following table lists the detailed description of the output files:

Name	Description
*.tlf	TLF Model
*.lib	Synopsys Model
*.net	Cdl netlist
*.gds	GDS file
*.ds	Datasheet
*.v	Verilog Model
*.vhdl	VHDL Model
*.lef	LEF view
*_antenna.lef	Antenna LEF view

*_antenna.clf	Antenna CLF model

And click on *Exit* button to quit the SRAM compiler.